目前分离跟android无关的模块：

添加单个会员数据

批量添加会员数据

编辑会员数据

删除会员数据(包括批量)

查找会员数据(分页)

添加AccessLog

删除AccessLog：客户端删除，数据库不删除，数据库添加标识

查找AccessLog(分页)

以下为公共部分的工具类定义，方法暂未实现，实现的目标是能够对sqlite进行高效率的增删改查。

*/\*\*  
 \* huangzhijin  
 \* 添加单个会员  
 \* 返回0：成功  
 \* 返回1：失败  
 \*/*public static Integer addMember(Member member){  
 return 0;  
}  
  
*/\*\*  
 \* 会员添加测试用例  
 \* 查看数据库是否成功添加数据  
 \*/*public static void addMemberTest(){  
 Member member=new Member();  
 member.setName("huangzhijin");  
 member.setType(0);  
 member.setFeature("");  
 byte[] bytes=new byte[100];  
 member.setImage(bytes);  
 member.setGreeting("hello huzngzhijin");  
 *addMember*(member);  
}  
  
*/\*\*  
 \* huangzhijin  
 \* 批量添加会员  
 \* 返回0：成功  
 \* 返回1：失败  
 \*/*public static Integer addMemberBatch(List<Member> memberList){  
 return 0;  
}  
  
*/\*\*  
 \*  
 \* 批量添加会员测试用例  
 \* 查看数据库是否成功添加数据  
 \*/*public static void addMemberBatchTest(){  
 List<Member> memberList=new ArrayList<Member>();  
 for(int i=0;i<50;i++){  
 Member member=new Member();  
 member.setName("member"+i);  
 member.setType(0);  
 member.setFeature("");  
 byte[] bytes=new byte[100];  
 member.setImage(bytes);  
 member.setGreeting("hello member"+i);  
 memberList.add(member);  
 }  
 *addMemberBatch*(memberList);  
}

*/\*\*  
 \* huangzhijin  
 \* 入参：path是xlsx表的文件路径  
 \* 批量添加会员  
 \* 返回0：成功  
 \* 返回1：失败  
 \*/*public static Integer addMemberBatch(String path){  
 return 0;  
}  
  
*/\*\*  
 \*  
 \* 导入excel表方式批量添加会员测试用例  
 \* 查看数据库是否成功添加数据  
 \*/*public static void addMemberBatchTest2() {  
 String path="D:\\test.xlsx";  
 *addMemberBatch*(path);  
}  
  
  
*/\*\*  
 \* huangzhijin  
 \* 编辑会员  
 \* 返回0：成功  
 \* 返回1：失败  
 \*/*public static Integer editMember(Member member){  
 return 0;  
}  
  
*/\*\*  
 \* 修改会员测试用例  
 \* 查看数据库会员数据是否修改成功  
 \*/*public static void editMemberTest(){  
 Member member=new Member();  
 member.setId(1);  
 member.setName("huangzhijin");  
 member.setType(0);  
 member.setFeature("");  
 byte[] bytes=new byte[100];  
 member.setImage(bytes);  
 member.setGreeting("hello huzngzhijin");  
 *editMember*(member);  
}  
  
*/\*\*  
 \* huangzhijin  
 \* 删除会员，可删除1个或多个  
 \* 返回0：成功  
 \* 返回1：失败  
 \*/*public static Integer deleteMembers(Integer[] ids){  
 return 0;  
}  
  
  
*/\*\*  
 \* huangzhijin  
 \* 删除会员测试用例  
 \* 查看数据在数据库中是否删除成功  
 \*/*public static void deleteMembersTest(){  
 Integer[] ids=new Integer[10];  
 ids[0]=1;  
 ids[1]=2;  
 *deleteMembers*(ids);  
}  
  
*/\*\*  
 \* huangzhijin  
 \* 根据id查找会员数据  
 \*  
 \*  
 \*/*public static Map<String,Object> getMemberById(Integer id){  
 return null;  
}  
  
*/\*\*  
 \* huangzhijin  
 \* 根据id查找会员数据测试用例  
 \* 参看是否拿到相应的数据  
 \*/*public static void getMemberByIdTest(){  
 Integer id=1;  
 Map<String,Object> map=*getMemberById*(id);  
 String name=map.get("name").toString();  
 String type=map.get("type").toString();  
 String image=map.get("image").toString();  
 String feature=map.get("feature").toString();  
 String greeting=map.get("greeting").toString();  
 System.*out*.println("name:"+name);  
 System.*out*.println("type:"+type);  
 System.*out*.println("image:"+image);  
 System.*out*.println("feature:"+feature);  
 System.*out*.println("greeting:"+greeting);  
}  
  
*/\*\*  
 \* huangzhijin  
 \* 会员数据条件查询,比如根据name或type进行查询  
 \*/*public static List<Map<String,Object>> getMembers(Member member){  
 return null;  
}  
  
*/\*\*  
 \* huangzhijin  
 \* 会员数据条件查询测试用例  
 \*/*public static void getMembersTest(){  
 Member member=new Member();  
 member.setName("huangzhijin");  
 List<Map<String,Object>> list=*getMembers*(member);  
 for(int i=0;i<list.size();i++){  
 Map<String,Object> map=list.get(i);  
 String id=map.get("id").toString();  
 String name=map.get("name").toString();  
 String type=map.get("type").toString();  
 String image=map.get("image").toString();  
 String feature=map.get("feature").toString();  
 String greeting=map.get("greeting").toString();  
 System.*out*.println("id:"+id);  
 System.*out*.println("name:"+name);  
 System.*out*.println("type:"+type);  
 System.*out*.println("image:"+image);  
 System.*out*.println("feature:"+feature);  
 System.*out*.println("greeting:"+greeting);  
 }  
}  
  
*/\*\*  
 \* huangzhijin  
 \* 添加AccessLog  
 \* 返回0：成功  
 \* 返回1：失败  
 \*/*public static Integer addAccessLog(AccessLog accessLog){  
 return 0;  
}  
  
*/\*\*  
 \* huangzhijin  
 \* 添加AccessLog测试用例  
 \* 查看数据库是否添加成功  
 \*/*public static void addAccessLogTest(){  
 AccessLog accessLog=new AccessLog();  
 accessLog.setName("huangzhijin");  
 byte[] image=new byte[100];  
 accessLog.setImage(image);  
 accessLog.setTimestamp(1560307900);  
 *addAccessLog*(accessLog);  
}  
  
  
*/\*\*  
 \* huangzhijin  
 \* 删除AccessLog，可删除一条或者多条记录  
 \* 返回0：成功  
 \* 返回1：失败  
 \*/*public static Integer deleteAccessLogs(Integer[] ids){  
 return 0;  
}  
  
*/\*\*  
 \* huangzhijin  
 \* 删除AccessLog测试用例  
 \* 查看数据库是否成功删除数据  
 \*/*public static void deleteAccessLogsTest(){  
 Integer[] ids=new Integer[10];  
 ids[0]=1;  
 ids[2]=2;  
 *deleteAccessLogs*(ids);  
}  
  
*/\*\*  
 \* huangzhijin  
 \* 根据ID获取AccessLog  
 \*/*public static Map<String,Object> getAccessLogById(Integer id){  
 return null;  
}  
  
*/\*\*  
 \* huangzhijin  
 \*根据ID获取AccessLog测试用例  
 \*/*public static void getAccessLogByIdTest(){  
 Integer id=1;  
 Map<String,Object> map=*getAccessLogById*(id);  
 String name=map.get("name").toString();  
 String timestamp=map.get("timestamp").toString();  
 System.*out*.println("name:"+name);  
 System.*out*.println("timestamp:"+timestamp);  
}  
  
*/\*\*  
 \* huangzhijin  
 \* 获取AccessLog列表  
 \*/*public static List<Map<String,Object>> getAccessLogList(){  
 return null;  
}  
  
*/\*\*  
 \* huangzhijin  
 \* 获取AccessLog列表测试用例  
 \*/*public static void getAccessLogListTest(){  
 List<Map<String,Object>> list=*getAccessLogList*();  
 for(int i=0;i<list.size();i++){  
 Map<String,Object> map=list.get(i);  
 String name=map.get("name").toString();  
 String type=map.get("type").toString();  
 String timestamp=map.get("timestamp").toString();  
 System.*out*.println("name:"+name);  
 System.*out*.println("type:"+type);  
 System.*out*.println("timestamp:"+timestamp);  
 }  
}  
  
*/\*\*  
 \* huangzhijin  
 \* 根据类型查找AccessLog会员测试用例  
 \* type: 0->会员AccessLogs 1->非会员AccessLogs 2->卡式会员AccessLogs  
 \*/*public static List<Map<String,Object>> getAccessLogsByType(Integer type){  
 return null;  
}  
  
*/\*\*  
 \* huangzhijin  
 \* 根据类型查找AccessLog测试用例  
 \*/*public static void getAccessLogsByType(){  
 List<Map<String,Object>> list=*getAccessLogsByType*(0);  
 for(int i=0;i<list.size();i++){  
 Map<String,Object> map=list.get(i);  
 String name=map.get("name").toString();  
 String timestamp=map.get("timestamp").toString();  
 System.*out*.println("name:"+name);  
 System.*out*.println("timestamp:"+timestamp);  
 }  
}